



SPICY INVADER

ETML: BELKHIRIA SOFIENE



04 SEPTEMBRE 2023

ETML
Venne

Table des matières


DataBases	2
Dump MySQL	2
Requêtes de sélection :	3
Gestion Rôle :	8
POO	11
Introduction :	11
Analyse fonctionnelle :	11
Analyse Technique :	11
Tests Unitaire :	11
ChatGPT :	11
Conclusion :	11

DataBases

Dump MySQL :

1. La première étape est de se connecter à MySQL pour cela aller dans un le cmd où se trouve votre serveur MySQL et utiliser la commande suivante « `mysql -u root -p` ».
 - a. « `mysql` » indique simplement que la commande s'adresse directement au client MySQL.
 - b. « `-u root` » désigne quel utilisateur se connecte au serveur dans ce cas l'utilisateur est « `root` » (Utilisateur par défaut)
 - c. « `-p` » sert à indiquer à MySQL qu'un mot de passe est attendu, le mot de passe de `root` dans ce cas-là, une fois la commande entrée MySQL vous proposera donc d'entrer votre mot de passe il est aussi possible d'écrire le mot de passe de l'utilisateur directement après le « `-p` » mais cela n'est pas conseillé pour cause de sécurité, il est aussi possible que lors d'un mot de passe écrit directement après « `-p` » MySQL vous retourne un Warning mais la commande sera tout de même exécutée.
 - d. Une fois ces commandes faites-vous devriez être dans MySQL votre

terminal devrait donc être comme ceci :




2. Une fois la première étape effectuée il est temps de dump la base de données pour se faire exécuter la commande ci-dessous :



```
mysql -uroot -proot < D:\PROJETS\spicy_invader\db_space_invaders.sql
```

- a. « `<` » se signe indique à MySQL que les informations d'un fichier vont être transmises dans la base de données ou directement dans MySQL (à noter que le signe `>` indique que les informations d'une base de données vont être transmises à un fichier extérieur)
- b. « `D:\PROJETS\spicy_invader\db_space_invaders.sql` » Cette partie est tout simplement le chemin du fichier il vous faudra donc le changer par l'endroit où votre fichier « `.sql` » se trouve
- c. Une fois cette commande effectuée vous pouvez vérifier que la base de données a bien été ajoutée en tapant la commande « `SHOW DATABASES ;` » qui a pour but de montrer les bases de données se trouvant sur le serveur.

3. Dans le cas ou votre serveur MySQL se trouve dans un conteneur docker il est préférable d'ouvrir un nouveau CMD dans un répertoire autre que celui du docker et d'exécuter la commande ci-dessous :

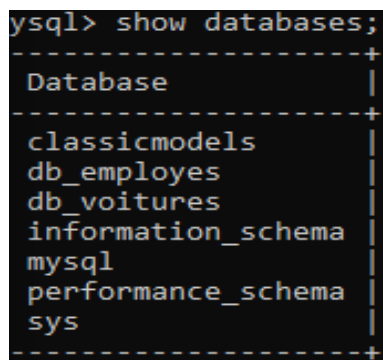


```
docker exec -i mysql-container-id mysql -uroot -proot < db_space_invaders.sql
```

- a. « docker exec-i » sers à exécuter quelque chose dans un conteneur docker c'est pour cela que après le «i » il vous est demander d'introduire l'identifiant du conteneur que vous souhaitez utiliser.
- b. A noté que « -i » indique que à la suite de l'exécution de la commande vous aller sortir du conteneur à l'opposé de « -it » qui lui indique que vous allez rester dans le conteneur à la suite de l'exécution de la commande.
- c. Le reste de la commande est la même qu'expliqué ci-dessus.
- d. En résumé la commande rentre dans un conteneur docker précis, exécute une commande MySQL et ressort juste après.

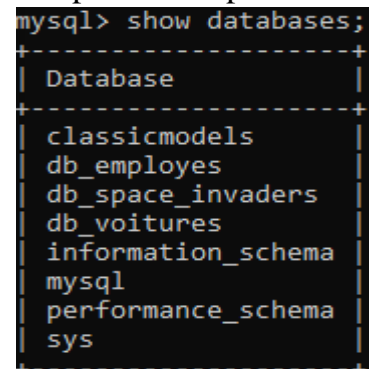
4. Voici un exemple de ce qui devrait se passer une fois les étapes effectuées.

- a. Avant les étapes :



```
mysql> show databases;
+-----+
| Database |
+-----+
| classicmodels |
| db_employes   |
| db_voitures   |
| information_schema |
| mysql         |
| performance_schema |
| sys           |
+-----+
```

- b. Après les étapes



```
mysql> show databases;
+-----+
| Database |
+-----+
| classicmodels |
| db_employes   |
| db_space_invaders |
| db_voitures   |
| information_schema |
| mysql         |
| performance_schema |
| sys           |
+-----+
```

Requêtes de sélection :

Sélection 1 :

- Le but de cette sélection est de sélectionner les 5 meilleurs joueurs en termes de point de space invaders.
 - o Pour se faire il faut exécuter la commande suivante :



```
SELECT * FROM `t_joueur` ORDER BY jouNombrePoints DESC limit 5;
```

- La commande « SELECT » sert à indiquer à MySQL quelle est la liste des colonnes à sélectionner à noter que « * » permet de sélectionner toutes les colonnes d'une table
- La commande « FROM » sert à indiquer depuis quelle table la colonne doit être prise il s'agit de la colonne « t_joueur » dans notre cas.
- « ORDER BY » indique par quelle table nous voulons ordonner dans notre cas nous allons ordonner par le nombre de points de chaque joueur, à noter que « ORDER BY » ordonne par défaut du plus petit au plus grand.
- La spécification « DESC » se place après un « ORDER BY » afin d'inverser le sens de l'ordre, pour donc le faire passer du plus grand au plus petit.
- La spécification « LIMIT » sert à instaurer une limite au nombre de ligne afficher, dans notre cas, nous voulons ne voir que les 5 premier du classement c'est pour cela que nous limitant la sélection au 5 première lignes.

Sélection 2 :

- Le but de cette sélection est de Trouver le prix maximum, minimum et moyen des armes. Les colonnes doivent avoir pour nom « PrixMaximum », « PrixMinimum » et « PrixMoyen »
 - Pour se faire il faut exécuter la commande suivante :

```
SELECT MAX(armPrix) as PrixMaximum, MIN(armPrix) as PrixMinimum, AVG(armPrix) as PrixMoyen FROM t_arme;
```

- « MAX () » sert à sélectionner la ligne avec la valeur la plus haute (maximum) de la colonne (colonne se trouvant dans les parenthèses).
- « AS » sert à renommer une colonne afin de simplifier la lecture de l'affichage une fois la sélection terminée, dans notre cas, nous renomons la colonne « MAX(armPrix) » en PrixMaximum.
- « MIN () » sert à sélectionner la ligne avec la valeur la plus basse (minimum) de la colonne (colonne se trouvant dans les parenthèses).
- « AVG() » sert à effectuer la moyenne de toute la colonne (colonne se trouvant dans les parenthèses) dans notre cas nous faisons la moyenne du prix des arme.

Sélection 3 :

- Le but de cette sélection est de Trouver le nombre total de commandes par joueur et trier du plus grand nombre au plus petit. La 1ère colonne aura pour nom "IdJoueur", la 2ème colonne aura pour nom "NombreCommandes".

- Pour se faire il faut exécuter la commande suivante :

```
SELECT fkJoueur as IdJoueur, count(fkJoueur) AS NombreCommande FROM t_commande GROUP BY
fkJoueur ORDER BY NombreCommande DESC;
```

- « COUNT() » sert à compter le nombre de ligne d'une colonne (colonne se trouvant dans les parenthèses) dans notre cas nous comptons les lignes de la colonne « fkJoueur »
- La fonction « GROUP BY » permet de grouper les résultats d'une sélection en fonction d'une colonne dans notre cas nous voulons afficher un résultat pour chaque « fkJoueur » c'est pour cela que nous groupons en fonction de « fkJoueur » à noter que vous ne pouvez que grouper par une colonne qui se trouve dans le select.

Sélection 4 :

- Le but de cette sélection est de trouver les joueurs qui ont passé plus de 2 commandes. La 1ère colonne aura pour nom "IdJoueur", la 2ème colonne aura pour nom "NombreCommandes".
 - Pour se faire il faut exécuter la commande suivante :

```
SELECT fkJoueur as IdJoueur, count(fkJoueur) AS NombreCommande FROM t_commande GROUP BY
fkJoueur HAVING NombreCommande > 2;
```

- La clause « HAVING » permet d'insérer une condition au tris suite à un GROUP BY dans notre cas nous conditionnons la sélection pour n'avoir que les gens qui ont un nombre de commande plus grand que deux.

Sélection 5 :

- Le but de cette sélection est de trouver le pseudo et le nom de l'arme pour chaque commande.
 - Pour se faire il faut exécuter la commande suivante

```
SELECT t_commande.idCommande, t_joueur.jouPseudo, t_arme.armNom
FROM t_detail_commande JOIN t_arme ON idArme =
t_detail_commande.fkArme JOIN t_commande ON
t_detail_commande.fkCommande = t_commande.idCommande JOIN
t_joueur ON t_commande.fkJoueur = t_joueur.idJoueur;
```

- Le mot clef « JOIN » ou « INNER JOIN » permet de faire une sélection sur plusieurs tables, on se sert donc de join pour importer une nouvelle table dans la sélection, dans ce cas nous importons les tables arme, commande et joueur.

- Le mot clef « ON » s'utilise avec le « JOIN », effectivement un « JOIN » ne peut s'utiliser sans « ON », celui-ci sert à indiquer un point d'attache entre les table, il est donc pas possible d'effectuer un « JOIN » sans « ON » car cela impliquerait de l'effectuer sans point d'attache, pour trouver un point d'attache il suffit de regarder un MLD de la base et de trouver dans une table une colonne contenant une valeur rapportée d'une autre.
- Exemple : dans la commande ci-dessus la colonne « fkARme » de la table détail commande est liée à la colonne « idArme » de la table arme ce qui veut dire que leurs valeurs sont identiques c'est pour cela que l'on peut s'en servir comme lien entre les deux tables.

Sélection 6 :

- Le but de cette sélection est de Trouver le total dépensé par chaque joueur en ordonnant par le montant le plus élevé en premier, et limiter aux 10 premiers joueurs. La 1ère colonne doit avoir pour nom "IdJoueur" et la 2ème colonne "TotalDepense"
 - Pour cela nous devons faire exécuter cette commande :

```
SELECT t_joueur.idJoueur AS IDjOUEUR, SUM(t_arme.armPrix *
t_detail_commande.detQuantiteCommande) AS totalDepense FROM
t_detail_commande JOIN t_arme ON idArme =
t_detail_commande.fkArme JOIN t_commande ON
t_detail_commande.fkCommande = t_commande.idCommande JOIN
t_joueur ON t_commande.fkJoueur = t_joueur.idJoueur GROUP BY
t_joueur.idJoueur ORDER BY totalDepense DESC LIMIT 10;
```

Sélection 7 :

- Le but de cette sélection est de Récupérez tous les joueurs et leurs commandes, même s'ils n'ont pas passé de commande. Dans cet exemple, même si un joueur n'a jamais passé de commande, il sera quand même listé, avec des valeurs `NULL` pour les champs de la table `t_commande`.
 - Pour cela nous devons taper cette commande :

```
SELECT jouPseudo, t_commande.comNumeroCommande FROM t_joueur LEFT
JOIN t_commande ON t_commande.fkJoueur = idJoueur;
```

- L'attribut « LEFT JOIN » sers tout comme le join à joindre une table cependant cette fois ci nous allons prendre toutes les infos de la table de

gauche (table étant avant le left) ce qui veut dire que nous prenons en compte les champs null étant donné que nous prenons tous les champs.

Sélection 8.

- Le but de cette sélection est de récupérer toutes les commandes et afficher le pseudo du joueur si elle existe, sinon montrer `NULL` pour le pseudo.
 - Pour ce faire nous exécutons cette commande :

```
SELECT jouPseudo, t_commande.comNumeroCommande
FROM t_joueur
RIGHT JOIN t_commande ON t_commande.fkJoueur = idJoueur;
```

- L'attribut « RIGHT JOIN » agit en tout point comme le « LEFT », cependant nous prenons tous les enregistrements de la table de droite (table après le right) et non de gauche.

Sélection 9.

- Le but de cette sélection est de trouver le nombre total d'armes achetées par chaque joueur (même si ce joueur n'a acheté aucune Arme)
 - Pour ce faire nous tapons cette commande :

```
SELECT jouPseudo, SUM(t_detail_commande.detQuantiteCommande) AS
nombreArme FROM t_detail_commande JOIN t_arme ON t_arme.idArme =
t_detail_commande.fkArme JOIN t_commande ON t_commande.idCommande
= fkCommande RIGHT JOIN t_joueur ON t_joueur.idJoueur =
t_commande.fkJoueur GROUP BY jouPseudo;
```

Sélection 10 :

- Le but de cette sélection est de Trouver les joueurs qui ont acheté plus de 3 types d'armes différentes :

```
SELECT t_joueur.jouPseudo, COUNT(t_detail_commande.fkArme) AS TotalArmeAchete FROM
t_detail_commande
LEFT JOIN t_commande ON t_commande.idCommande = t_detail_commande.fkCommande
RIGHT JOIN t_joueur ON t_commande.fkJoueur = t_joueur.idJoueur
GROUP BY t_joueur.idJoueur
HAVING COUNT(t_detail_commande.fkArme) > 3;
```


Gestion Rôle :

But :

- Le but de cette partie est de montrer comment créer des utilisateurs et des rôles pouvant interagir avec la base de données.
- Selon les bases de données on pourrait être amené à créer un utilisateur pour chaque situation cependant si nous devons gérer des milliers d'utilisateurs il est plus efficace de créer des rôles ayant directement les privilèges voulus et leur attribuer directement ces rôles aux utilisateurs.


Administrateur :

- Pour créer le rôle administrateur nous devons utiliser la commande suivante :




```
CREATE ROLE 'r_Administrateur';
```

- Une fois le rôle créé nous devons lui attribuer les privilèges souhaités, dans notre cas il nous a demandé de donner « CRUD » qui correspond aux rôles « CREATE », « READ (SELECT dans notre cas) », « UPDATE » et « DELETE »
- Pour ce faire nous devons exécuter la commande suivante :



```
GRANT CREATE, SELECT, UPDATE, DELETE ON *.*  
TO 'r_Administrateur';
```

- En plus de CRUD il nous a demandé de rajouter l'option de gestion d'utilisateur, pour ce faire nous devons faire cela :



```
GRANT GRANT OPTION ON *.* TO  
'r_Administrateur';
```

Joueur

- Pour créer le rôle Joueur nous devons utiliser la commande suivante :

```
CREATE ROLE 'r_Joueur';
```

- Une fois le rôle créé nous devons lui attribuer les privilèges souhaités, dans notre cas il nous faut donner la possibilité de lire les informations des armes, il nous faut donc exécuter cette commande :

```
GRANT SELECT ON  
db_space_invaders.t_arme TO 'r_Joueur';
```

- Il doit aussi pouvoir créer une commande, pour cela exécuter cette commande :

```
GRANT CREATE ON  
db_space_invaders.t_commande TO  
'r_Joueur';
```

- Puis pour finir il doit aussi être capable de lire les commandes, pour ce faire exécuter cela :

```
GRANT SELECT ON  
db_space_invaders.t_commande TO  
'r_Joueur';
```

Gestionnaire de la boutique :

- Pour créer le rôle Gestionnaire de boutique nous devons la commande suivante :

```
CREATE ROLE  
'r_GestionnaireBoutique';
```

- Il doit pouvoir mettre à jour, lire et supprimer des armes :



```
GRANT UPDATE, SELECT, DELETE,  
INSERT ON  
db_space_invaders.t_arme TO  
'r_GestionnaireBoutique';
```

- Il doit pouvoir lire toutes les commandes :



```
GRANT SELECT ON  
db_space_invaders.t_commande TO  
'r_GestionnaireBoutique';
```

-

POO

Introduction :

Analyse fonctionnelle :

Analyse Technique :

Diagramme de classe :

Explication :

Tests Unitaire :

ChatGPT :

Conclusion :